In the Claims:

1.   (Currently amended) A processor executing a plurality of instructions, comprising:

an arithmetic logic unit (ALU); and

a plurality of registers coupled to the ALU, each register programmable to store a register value;

wherein said processor executes a routine having a test and skip instruction defined by an opcode, said test and skip instruction includes an immediate value and a register reference control bit contained within a source operand of the test and skip instruction, the test and skip instruction performs a comparison using the immediate value and the register value stored in the register referenced by the register reference control bit, and selectively skips a subsequent instruction that follows the test and skip instruction based on the comparison;

wherein the register reference control bit specifies whether the register reference is to a register from a first group of registers or to a register from a second group of registers, and if a register from the first group of registers is specified by said register reference control bit, the comparison is performed by comparing the immediate value to the register value, and, if a register from the second group of registers is specified by said at least one register reference control bit, the comparison is performed by masking the register value with the immediate value and examining one or more bits in [[the]] a masked version of the referenced register; and

wherein the subsequent instruction jumps to another routine.

2.   (Canceled).

3.   (Previously presented) The processor of claim 1 wherein, if comparing the immediate value to the register value, the processor skips the subsequent instruction if the immediate value does not match the register value and executes the subsequent if the immediate value does match the register value.

4.   (Canceled).

5.   (Previously Presented) The processor of claim 1 wherein, if masking the register value, the masking is performed by ANDing the immediate value with the register value.

6-7.   (Canceled).

8.   (Currently amended) The processor of claim 1 wherein the <u>plurality of</u> registers include a status register and if the register reference specified by said at least one bit is not the status register, the comparison is performed by comparing the immediate value to the register value in the referenced register, and, if the register reference specified by said at least one bit is the status register, the comparison is performed by masking the register value in the status register with the immediate value and examining one or more bits in [[the]] <u>a</u> masked version of the status register.

9. (Currently amended) A method of executing an instruction defined by an opcode, an immediate value, and a register reference control bit ~~contained within a source operand of the instruction~~ that dictates one of at least two tests, the method comprising:

examining said register reference control bit <u>contained within a source operand of the instruction</u> to determine its state;

if said register reference control bit is in a first state, comparing the immediate value to the contents of a register in a first group of registers and skipping a subsequent instruction based on [[the]] <u>an</u> outcome of the comparison; or

if said register reference control bit is in a second state, masking the contents of a register in [[the]] <u>a</u> second group of registers with the immediate value, testing one or more bits in the masked version of the contents of the register, and skipping a subsequent instruction based on [[the]] <u>an</u> outcome of the testing; and

wherein said register reference control bit is outside said opcode.

10. (Original) The method of claim 9 wherein skipping the subsequent instruction comprises replacing the subsequent instruction with a no operation instruction.

11. (Currently amended) A system, comprising:

a main processor unit; and

a co-processor coupled to said main processor unit, wherein said co-processor selectively operates in a stack-based instruction mode and a register-based instruction mode,

wherein, during the register-based instruction mode, the co-processor executes an instruction defined by an opcode and including an immediate value and a reference to a register accessible to said co-processor, performs a comparison using the immediate value and [[the]] a register value, and executes or skips a subsequent instruction based on the comparison; and

wherein the instruction includes at least one register reference control bit contained within a source operand of the instruction, separate from said opcode, that specifies whether the register reference is to a register from a first group of registers or to a register from a second group of registers, and if a register from the first group of registers is specified by said at least one register reference control bit, the comparison is performed by comparing the immediate value to the register value, and, if a register from the second group of registers is specified by said at least one register reference control bit, the comparison is performed by masking the register value with the immediate value and examining one or more bits in [[the]] a masked version of the referenced register.

12. (Canceled).

13. (Previously presented) The system of claim 11 wherein, if comparing the immediate value to the register value, the co-processor does not execute the subsequent instruction if the immediate value does not match the register

value and executes the subsequent if the immediate value does match the register value.

14.   (Canceled).

15.   (Previously presented) The system of claim 11 wherein, if masking the register value, the masking is performed by ANDing the immediate value with the register value.

16-17.   (Canceled).

18.   (Original) The system of claim 11 further comprising wireless communication circuitry and said system comprises a cell phone.

19. (Currently amended) A programmable logic device comprising;

control logic;

means for selectively changing an operating mode of the programmable logic device; and

means for decoding a register reference control bit contained within a source operand in an instruction that includes an immediate value and a reference to a register, for performing a comparison using the immediate value and a register value stored in the referenced register, and for causing the [[processor]] programmable logic device to execute or not execute a subsequent instruction that follows the instruction based on the comparison;

wherein said register reference control bit selectively specifies whether the comparison is to be performed by comparing the immediate value to the register value or whether the comparison is to be performed by masking the register value with the immediate value and examining one or more bits in [[the]] a masked version of the referenced register; and wherein the subsequent instruction jumps to a routine associated with a particular operating mode;

wherein said register reference control bit is separate from an opcode that defines said instruction.

20. (Currently amended) The [[system]] programmable logic device of claim 19 including means for comparing the immediate value to the register value in the referenced register.

21. (Previously presented) The processor of claim 1 wherein, if masking the register value, the processor skips the subsequent instruction if the masked version of the register value comprises all logic high values or all

logic low values, and executes the subsequent instruction if the masked version comprises a mix of logic high and low values.

22.   (Previously presented) The system of claim 11 wherein, if masking the register value, the processor skips the subsequent instruction if the masked version of the register value comprises all logic high values or all logic low values, and executes the subsequent instruction if the masked version comprises a mix of logic high and low values.